

#### Answers to Questions

1. 10
2. b
3. Last-In-First-Out; and First-In-First-Out
4. False. It's the other way around.
5. b
6. It doesn't move at all.
7. 45
8. False. They take  $O(1)$  time.
- 9-

1. d
2. comparing and swapping (or copying)
3. False
4. a
5. False
6. b
7. False
8. three
9. Items with indices less than or equal to outer are sorted.
10. c
11. d
12. copies
13. b

5-b

6-a

أجابة سؤال:

How many times would you need to traverse a singly linked list to delete the item with the largest key?

الأجابة once, if the links include a previous reference

أجابة سؤال:

Which do you think would be a better choice to implement a stack: a singly linked list or an array?

الأجابةUsually, the list. They both do push() and pop() in O(1) time, but the list uses memory more efficiently.

### A) True or false:

- Data can be retrieved from the QUEUE in LIFO order (F)
- Data can be retrieved from the STACK in FIFO order (F)
- An algorithm is a procedure for solving a problem in terms of the actions to execute and the order in which these action execute (T)
- The binary search is faster than the sequential search. (T)
- In the selection sort, the largest keys accumulate on the left (F)
- In the selection sort, a number of items must be shifted to insert each item in its correctly sorted position (F)
- In the selection sort, the sorted items accumulate on the right. (F)
- In the bubble sort, higher values float to the right whereas lower values float to the left. (T)
- The new items are always added to the top of the queue. (F)
- The pop, push and peek operations have the same effect on the stack's size. (F)
- In the insertion sort, after an item is inserted, it will never be shifted to the left. (T)

### B)

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots\right) \quad \text{Or}$$

$$\pi = 4\left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots\right) \quad \text{Or}$$

$$\pi = 4\left(\frac{(-1)^0}{1} + \frac{(-1)^1}{3} + \frac{(-1)^2}{5} + \frac{(-1)^3}{7} + \frac{(-1)^4}{9} + \frac{(-1)^5}{11} + \dots\right)$$

نلاحظ أن المقام هو عبارة عن متوالية عددية متزايدة بمقدار 2 و الأساس =1.

أما البسط فهو الرقم 1- مرفوع لأس متغير يبدأ من الصفر ويزداد بمقدار 1. ومن المعروف في مجال المتسلسلات أنه للحصول على نتيجة ذات دقة عالية فيجب تجميع عدد كبير جداً من حدود المتسلسلة وقد استخدمنا هنا العدد 50000000. ويمكنك عزيزي القارئ تجربة أعداد أقل ومقارنة النتيجة مع قيمة  $\pi$  المتعارف عليها.

```
using System;
using System.Collections.Generic;
using System.Text;
namespace pi
{
    class Program
    {
        static void Main(string[] args)
        {
            double pi = 0;
            int i = 0;
            int j = 0;
            for (i = 1; i < 100000000; i=i+2)
            {
                pi = pi + 4 * (Math.Pow(-1, j) / i);
                j++;
            }
            Console.WriteLine("{0}", pi);
        }
    }
}
```

C)

$$6c = 24$$

$$15$$

$$30$$

$$\sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$$

أثبت أن :

$$\sum_{i=0}^{n-1} i = 0 + 1 + 2 + \dots + (n-2) + (n-1)$$

بكتابة الصيغة السابقة بشكل معكوس:

$$\sum_{i=0}^{n-1} i = (n-1) + (n-2) + \dots + 1 + 0$$

وبجمع المعادلتين:

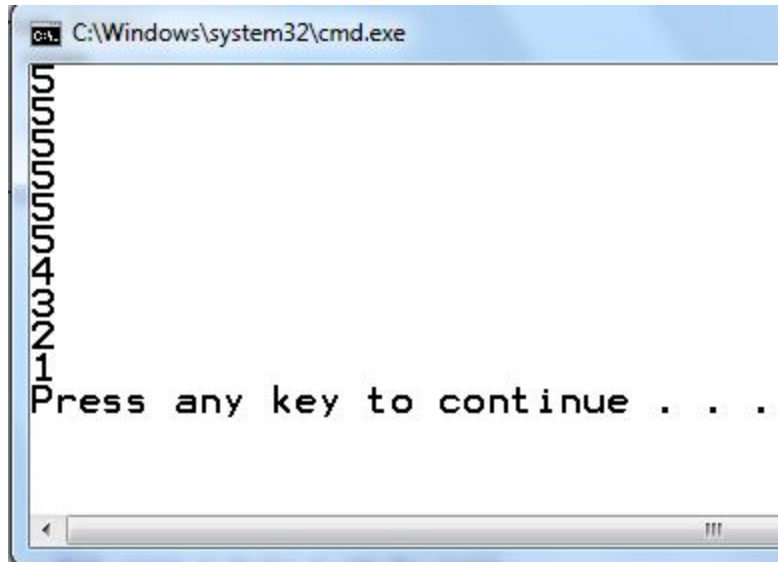
$$2 \times \sum_{i=0}^{n-1} i = (n-1) + (n-1) + \dots + (n-1) + (n-1) = n(n-1)$$

$$\sum_{i=0}^{n-1} i = n(n-1)/2$$

ومنها يمكن القسمة على 2

E) b

نتيجة تنفيذ البرنامج (السؤال الأخير)



```
C:\Windows\system32\cmd.exe
5
4
3
2
1
Press any key to continue . . .
```